Evaluation of NETWORK II.5

Julian J. Bunn

November 19th. 1990

Description of Product. NETWORK II.5 is a "general purpose computer and communications network performance evaluation tool". It is an event-driven, rather than an analogue, model. The emphasis is on timing, with many levels of modelling detail. It is supposed to be easy to modify a NETWORK II.5 simulation and re-run it. NETWORK II.5 is marketed by CACI Products Company, Hatboro, PA, USA.

Purpose of Evaluation. NETWORK II.5 was evaluated with regard to its suitability as a simulation of the SHIFT File and CPU Server configuration ("shift", version 1, CN Division, 25th. July, 1990).

Period of Evaluation. An evaluation kit was provided on June 20th. 1990. The evaluation period was from this date until November 16th. 1990. A course was attended, given by the author of NETWORK II.5 (W. Garrison), from 13th. to 16th. November. The estimated integrated time spent working with the product is four man-weeks.

Evaluation Platform. The product was evaluated on a VAXstation 2000 with 6 MB of main memory, running VMS 5.3 and DECwindows. During the course that was attended, use was made of the PC MS-DOS version, and no particular advantages or differences were noted, apart from improved speed. The NETWORK II.5 version was 5.03.

Installation. Installation was made from 6250 bpi tape. The installation procedure was straightforward.

People Involved. The evaluation was undertaken by J.J. Bunn/CN with help from B. Segal/CN.

Disclaimer. This evaluation represents the views of the author. It contains some subjective criticisms of the product, although an attempt has been made to be as objective as possible.

Summary. NETWORK II.5 is definitely worth investigating both as a tool for simulating simple WANs of processors or as a teaching aid. However, if the requirement is to simulate processors that execute complex job mixes, have multiple CPUs, or will be connected to a network switch (rather than Ethernet, Token Ring, FDDI etc.), such as the UltraNet "hub", then it is not recommended. In such cases alternatives such as the use of a suitable high-level language are recommended.

How it Works. NETWORK II.5 is written in the simulation language SIMSCRIPT II.5, which is marketed by the same company. It is provided as object-code only.

The model of the system to be simulated is defined in terms of three basic hardware elements, and four software elements. The hardware elements are "Processing Element" (PE), "Transfer Device" (TD), and "Storage Device" (SD). The software elements are basically "Instructions", "Modules", "Messages" and "Semaphores". PEs can execute one or more different Instructions, can send Messages over TDs to other PEs or SDs, and set/reset Semaphores, which are used as Global Variables. TDs can run one of several protocols (e.g. First Come First Served, Collision, Aloha, Token Ring, Slotted Token Ring). A Module is defined as a set of Instructions that can execute on one or more PEs. Messages are defined as having a name and a certain length. Semaphores can take a user-defined integer value. SDs have a specified capacity in bits, features such as read/write word overhead time, number of ports, and so on. Each PE has a "cycle time" which represents how long it takes to execute one Instruction. There are other simulation components available, not described here. There is a selection of provided distributions which may be used to define, for example, how often a module should be run, or how long

a message item is. It is the distributions which introduce the randomness into the model. User-defined distributions may also be specified.

The user specifies his model in terms of the above elements either by using a menu-driven tool, or a line-mode tool. These tools create a text file (e.g. SHIFT.NET) containing a description of the model. The content of the text file is, to a certain extent, self-describing. In practice using these tools turns out to be very laborious (particularly if one wants to make copies of PEs, SDs and TDs), and a text editor is better used to enter the .NET file directly. Once this is done, the tools need to be used to "Verify" the consistency of the proposed model.

This having been done, a different tool is invoked to read the .NET file and perform the simulation. The user must specify the simulation period, what level of reporting he wants (e.g. Snapshot, Event Trace etc.), what action he wishes to be taken on the occurence of an error, and so on. Running the simulation produces a text file showing, for example, how many times each PE was activated, which Modules it used, what messages were sent, the activity on each TD and SD. Another file is produced which may be used as input to a plotting tool, capable of showing the percentage utilization of each hardware element in the model, amongst other plots.

A complete record of what happened in the simulation is written to a binary file, which may be converted to normal text with a tool (provided). So a user-written analysis program could make its own displays from the simulation by decoding this file.

Quality of Documentation and On-Line Help. The product is provided with a 3 cm. thick user's manual. The manual describes in laborious detail how to operate the various component programs of NETWORK II.5. Most of these programs are designed to operate in a windowing environment, with pull-down menus etc.. The description of how to use these programs would be unnecessary, had there been good on-line help. But the only programs comprising on-line help were those orientated towards line-mode access. In the early stages of the evaluation, heavy reference had thus to be made to the manual.

The part of the manual devoted to describing the various simulation components was insufficiently detailed.

Ease of Use. It is easy to get started with NETWORK II.5. If the menu driven tools are used, then no programming is necessary. At the occurence of the first error message, the manual will almost certainly have to be consulted. As the user's model increases in complexity, his perception of the apparent richness of NETWORK II.5 is reduced, and he becomes more aware of its limitations. If the user is simulating an environment generally foreseen to be simulated by the authors of NETWORK II.5, then he will be well satisfied with the product. A typical target application would be the simulation of a system of insurance salesmen, sitting at remote terminals and interacting with a central database machine! As with any simulation, though, care must be taken when interpreting the results.

Plus Points. NETWORK II.5 is a good tool for quickly simulating a relatively simple system. It would make an excellent learning tool for studying WANs. The provided "Network Animation" tool gives a useful graphical real-time display of what is happening in the simulation.

Limitations. It is not possible easily to simulate a network hub, such as that provided with UltraNet. This is because a TD can serve only one PE at a time. Yet a hub provides concurrent use by several devices, up to a certain bandwidth limit.

Simulation of a multi-processor device is hard. In the Shift context, the simulation of, for example, an Apollo DN10000 four-processor machine running six jobs is essential. In NETWORK, this would require a PE for each CPU, a PE for Job Scheduling, a complex system of Semaphores for allocating the tasks to each CPU, collecting the data, and so on. Also, NETWORK II.5 disallows a named Module to concurrently execute with one or more copies on the same PE. So although time-slicing is available to allow equal priority modules to be moved in and out of a PE, those modules must be different modules.

Another limitation is that the names of all files that are to reside on SDs must be known in advance. There is no capability to generate a new file name during simulation. This would make simulation of staging extremely hard.

It is not possible to link a user-written SIMSCRIPT program with NETWORK II.5, which would be useful to do where limitations prevent the easy use of the standard constructs.

There is no easy way to randomise the choice of a PE for a particular Module to execute on.

Conclusion. The use of NETWORK II.5 is, in view of the above limitations, NOT recommended for the simulation of Shift. This is not to say that Shift *cannot* be simulated using this tool, but just that it would take a lot of effort, be very complex, and hard to interpret the results. A Shift simulation in NETWORK II.5 would be more difficult to upgrade and maintain than an equivalent program written in a high-level language. It is not clear that CERN could expect much response to requests for improvements to the features offered by NETWORK II.5, since their market is predominantly in the lucrative areas of banking and defence contracts.